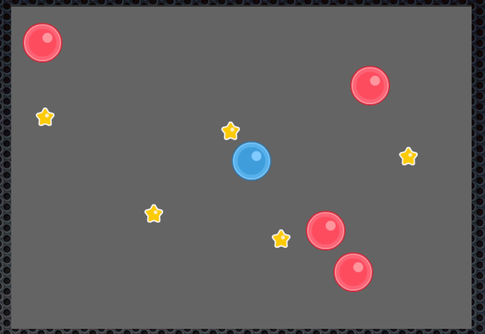
Entity Component System, Rust and Bevy

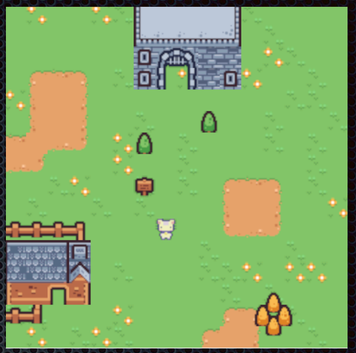
Bird’s-eye view

- Фулстек в monadical.com
- До: фриланс на Odesk/Upwork
- Адвокат ФП и статик типов

https://github.com/Firfi

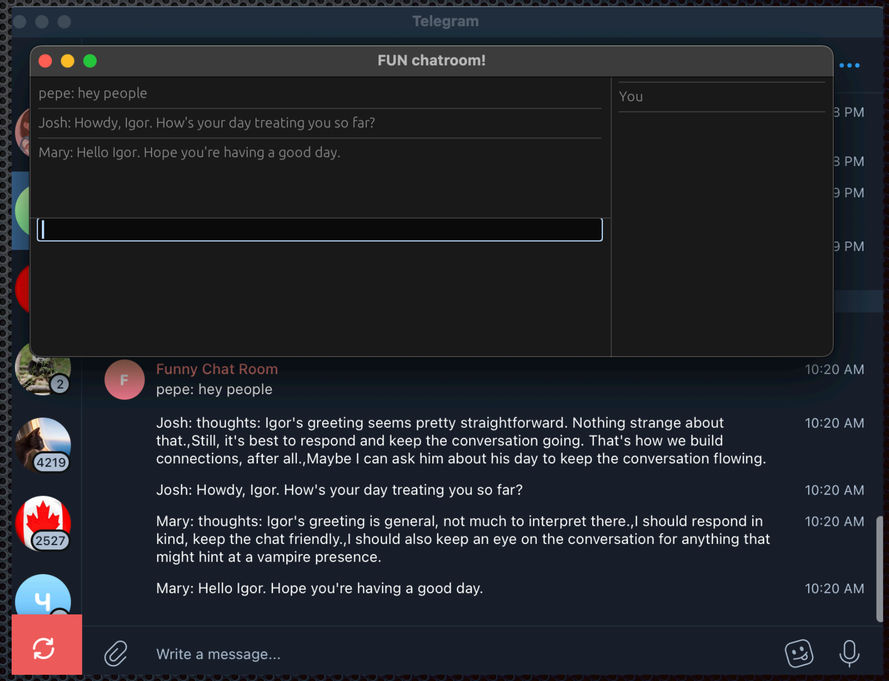
Игры-то пишешь?





(нет)

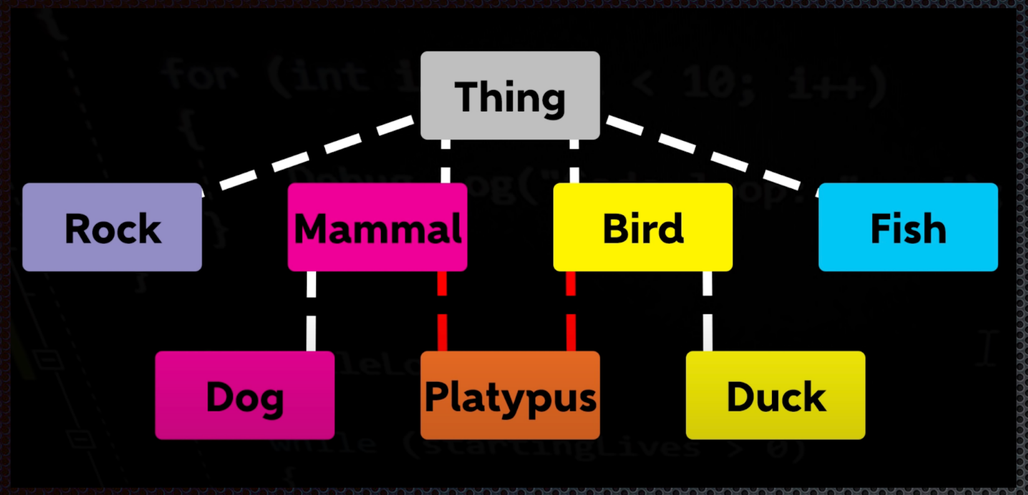
- Tutorials https://www.youtube.com/watch?v=TQt-v_bFdao&list=PLVnntJRoP85JHGX7rGDu6LaF3fmDDbqyd
- WebAssembly game-“website navigation”
- Websockets+chatgpt prompt engineering+telegram chatroom.exe



Entity Component System (ECS)

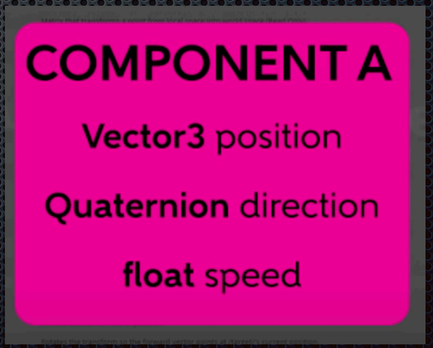
- Зачем это инженерам: кругозор / начитанность
- Зачем это не-инженерам: ???
- Что это: архитектурный паттерн
- Для чего: декаплинг симуляций в дата-ориентированном стиле
- А для чего: проблемы роста приложения

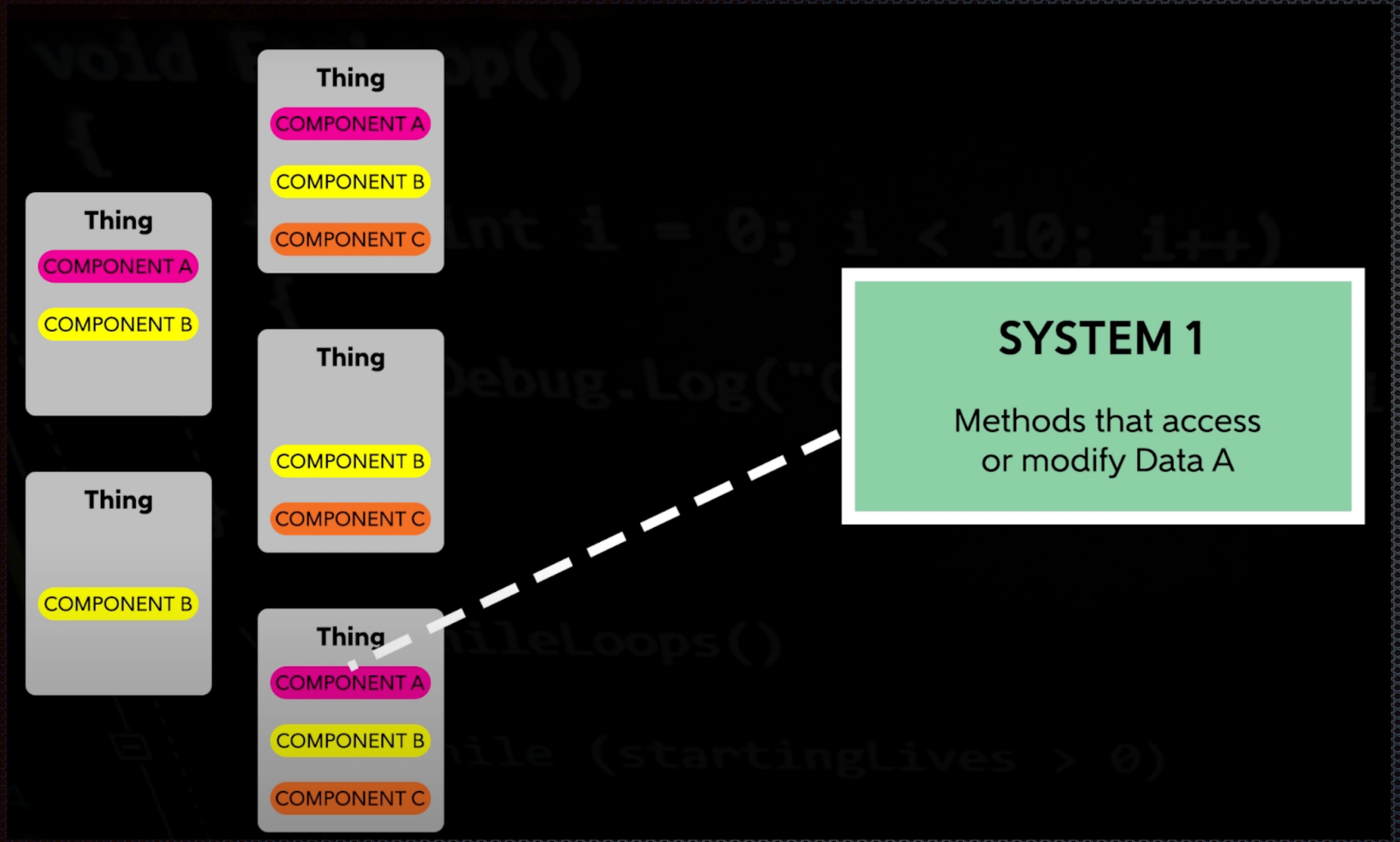
OOP


Thing
Rock
Mammal
Bird
Fish
Dog
Platypus
Duck

ECS


ENTITY
Thing
int hashID


COMPONENT A
Vector3 position
Quaternion direction
float speed



SYSTEM FOR ANYTHING THAT CAN MOVE

SYSTEM FOR ANYTHING THAT CAN BE SEEN

SYSTEM FOR ANYTHING THAT CAN TAKE DAMAGE

ECS

- Entities - “Identities”
- Components - “data” associated with Entities
- Systems - “behaviour” over data/identities

Entity Component Layout Table

Entity	Player	Enemy	Health	Power Up
0	x		x	
1		x	x	
2		x	x	
3				x
4				x
5		x	x	

ECS Data Model

Table: "players"

ID	has_sword	health	location
0	true	100	(0,0,0)
1	false	50	(1,1,1)
2	true	75	(1,2,3)
3	true	99	(3,2,1)
...	false	0	(-1,1,1)

Table: "monsters"

ID	has_sword	health	location
0	true	100	(0,0,0)
1	false	50	(1,1,1)
2	true	75	(1,2,3)
3	true	99	(3,2,1)
...	false	0	(-1,1,1)

VS

Table: "has_sword_AND_health_AND_location"

ID	has_sword	health	location
0	true	100	(0,0,0)
1	false	50	(1,1,1)
2	true	75	(1,2,3)
3	true	99	(3,2,1)
4	false	0	(-1,1,1)
5	true	100	(0,0,0)
6	false	50	(1,1,1)
7	true	75	(1,2,3)
8	true	99	(3,2,1)
...	false	0	(-1,1,1)

Simulation Loop and ECS

```
while (true) {
  ...game logic
}
```

→

```
while (true) {
    ...computation
    ...render
}
```

→

```
while (true) {
    system1()
    system2()
    system3()
}
```

```rust
struct GameState { ... }

fn main() {
    let mut game_state = initial_game_state();
    loop {
        let input_state = capture_input_state();

        input_system(&mut game_state, &mut input_state);

        ai_system(&mut game_state)
        physics_system(&mut game_state);

        // ...

        render_system(&mut game);
        audio_system(&mut game);

        wait_vsync();
    }
}
```

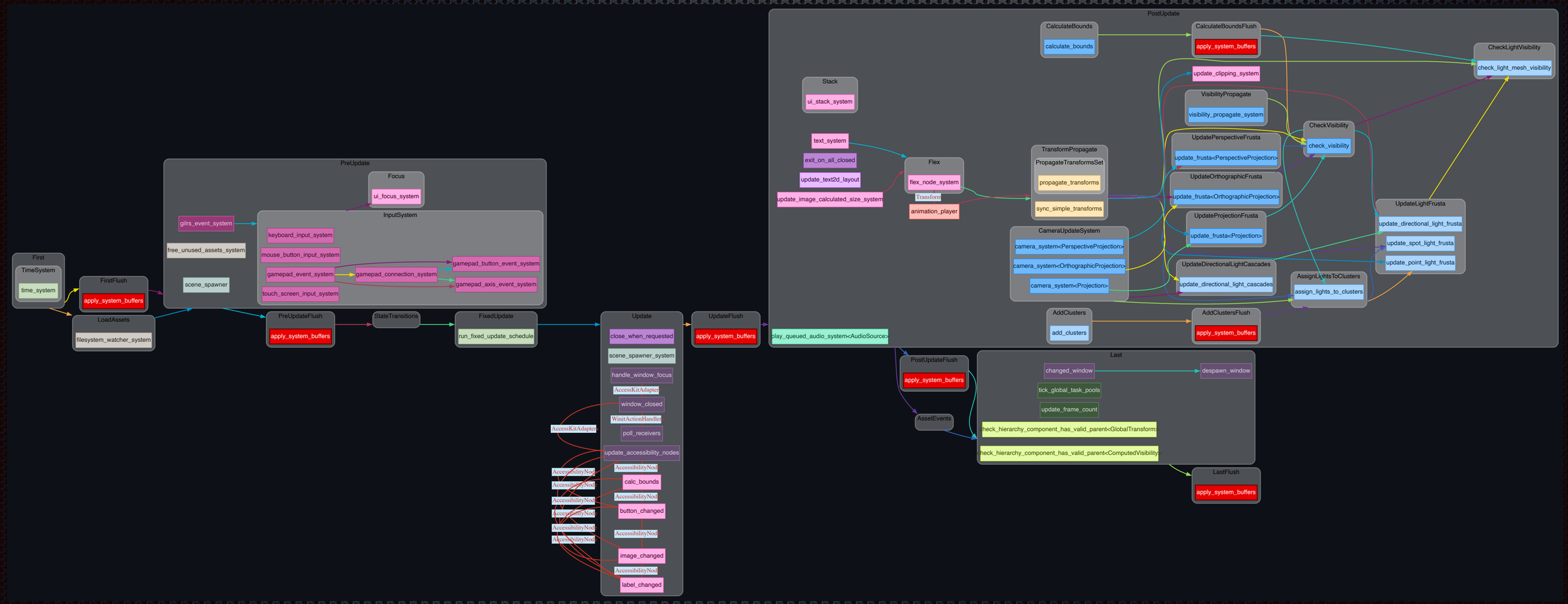
```rust
#[derive(Component)]
struct Position {
    x: f32,
    y: f32,
}

fn move_system(mut query: Query<&mut Position>) {
    for mut position :Mut<Position> in query.iter_mut() {
        position.x += 0.1;
        position.y += 0.1;
        println!("Updated position to: ({}, {})", position.x, position.y);
    }
}
```

No-loop example link

A *bevy* is a flock of birds. Although the term can be used for a variety of birds, it most often refers specifically to a *bevy* of quail. Other terms exist to describe flocks of other types of birds — a murder of crows, for example.

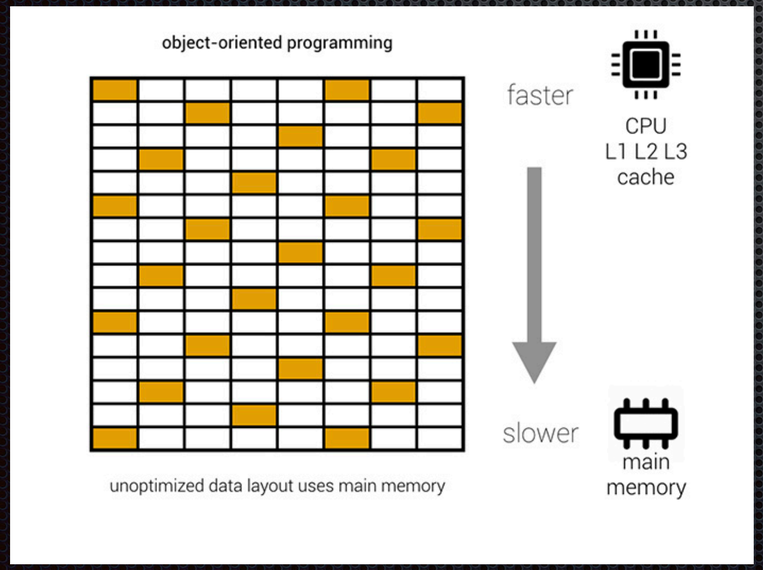
(Not ~~Twitter~~)

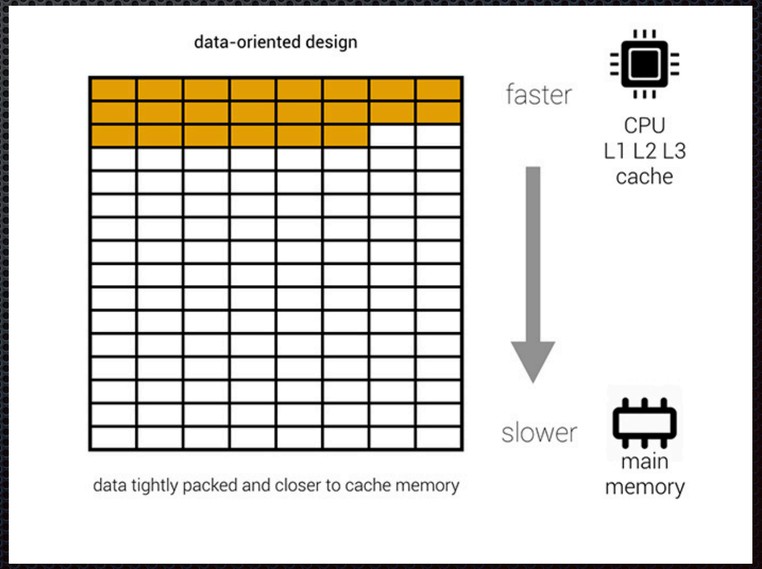
Loop Schedule (DAG!)



Why ECS in Bevy

- Their words: Data-oriented / Clean architecture / Performance
- My research: it’s also not viable to write classic OOP style in Rust (ref Chucklefish Games)
- Декларативно “какие данные мне нужны” позволяют оптимизацию кешей.





ECS outside of Rust

- Unity uses it for networking (сетевой код это очень сложно)
- Photon Quantum relies on it extensively
- The latter leverage ECS to help archive **computation determinism**
- Which is very important for networking optimization
- Intuition: compare with event sourcing

Links

https://www.youtube.com/watch?v=TQt-v_bFdao - basic explanation of ECS in Bevy, with nice “table-oriented” visualisation
https://devlog.hexops.com/2022/lets-build-ecs-part-2-databases/ - how data structures look like
https://github.com/recatek/gecs - One of standalone ECS implementations in Rust
https://www.kodeco.com/7630142-entity-component-system-for-unity-getting-started - unity
https://unity.com/solutions/real-time-multiplayer/network-transport-layer - where Unity utilizes it
https://ldtk.io - just a nice “CMS” for 2d tile-based game levels (from Dead Cells https://store.steampowered.com/app/588650/Dead_Cells/ creator, written in Zig)
https://www.youtube.com/watch?v=aKLntZcp27M - ECS in Rust by Catherine West from Chucklefish games (Starbound https://store.steampowered.com/app/211820/Starbound/)

Entity Component System Overview in 7 Minutes

https://www.youtube.com/watch?v=U03XXzcThGU - caves of QUD / sproggiwood